JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 10282881

(43)Date of publication of application: 23.10.1998

(51)Int.Cl.

G09C 1/00
G09C 1/00
G09C 1/00
H04L 9/08
H04L 9/30

(21)Application number: 09086302

(22)Date of filing: 04.04.1997

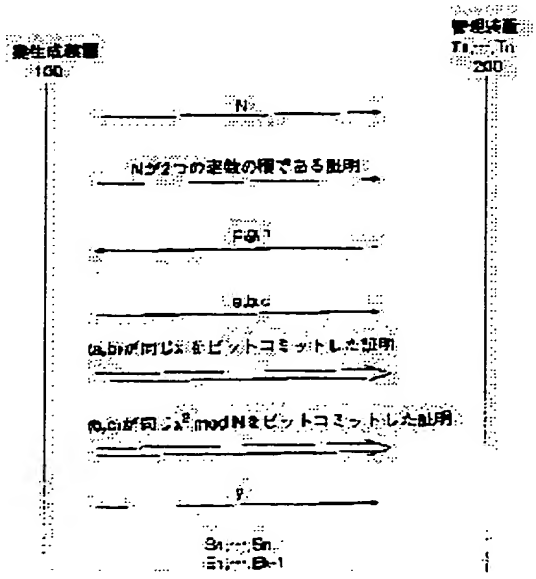
(71)Applicant: NIPPON TELEGR & TELEPH CORP <NTT>

(72)Inventor: OKAMOTO TATSUAKI

(54) SECRET KEY DECENTRALIZATION MANAGING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To restore a secret key by gathering a certain number of pieces of decentralized registration information for a prime factor factorization system open key cipher.
SOLUTION: A key generator K100 sends an open key N and information providing that N is the product of two prime numbers to an administrator Ti (i=1...n) 200, and T sends a prime number (p) and elements (g) and (h) whose order is N to K; and K generates random numbers (x) and r1, r2, and r3, calculates a=BCE (x, r1), b=BC (x, r2), and c=BCb (x2 mod N, r3) with a bit commitment function BC and sends them to Ti, and K prove that (a) and (b) made a bit commitment to the same (x) and (b) and (c) made a bit commitment to the same x2 mod N by exchanging



with T+i, sends (y) satisfying y2=(mod N) (y/N)=-1 to Ti, disperses 8x) to s1...sn and sends si to corresponding Ti, thereby sending pieces of information E1...Ek-1 providing that si is a correct value.

LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

E5353

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－282881

(43)公開日　平成10年(1998)10月23日

(51)Int.Cl.[6]		識別記号	FI		
Ｇ０９Ｃ	1/00	６３０	Ｇ０９Ｃ	1/00	６３０Ｄ
		６２０			６２０Ｂ
		６５０			６５０Ｚ
Ｈ０４Ｌ	9/08		Ｈ０４Ｌ	9/00	６０１Ｄ
	9/30				６６３Ｂ

審査請求　未請求　請求項の数２　ＯＬ　（全７頁）

(21)出願番号　特願平9－66302

(22)出願日　平成９年(1997)４月４日

(71)出願人　000004226
日本電信電話株式会社
東京都新宿区西新宿三丁目19番２号

(72)発明者　岡本　龍明
東京都新宿区西新宿三丁目19番２号　日本電信電話株式会社内

(74)代理人　弁理士　草野　卓

(54)【発明の名称】　秘密鍵分散管理方法

(57)【要約】

【課題】　素因数分解系公開鍵暗号に対し、分散登録情報をある一定数集めることで秘密鍵を復元可能とする。

【解決手段】　鍵生成者Ｋは公開鍵Ｎと、Ｎが２つの素数の積であることを証明する情報を管理者T_i（ｉ＝１，…，ｎ）へ送り、T_iは素数ｐと位数がＮの元ｇ，ｈをＫへ送り、Ｋは乱数ｘとr_1，r_2，r_3を生成し、ビットコミットメント関数ＢＣでａ＝BC_g（ｘ，r_1），ｂ＝BC_a（ｘ，r_2），ｃ＝BC_b（x^2 mod Ｎ，r_3）を演算してT_iへ送り、ＫはT_iとのやりとりでａとｂが同じｘをビットコミットしたもの、またｂとｃが同じx^2 mod Ｎをビットコミットしたものであることを証明し、$y^2 \equiv x^4 \pmod N$（ｙ／Ｎ）＝－１を満すｙをT_iへ送り、ｘをs_1,…，s_nに分散してs_iを対応T_iへ送りまたs_iが正しい値であることを証明する情報E_1,…，E_{k-1}を送る。

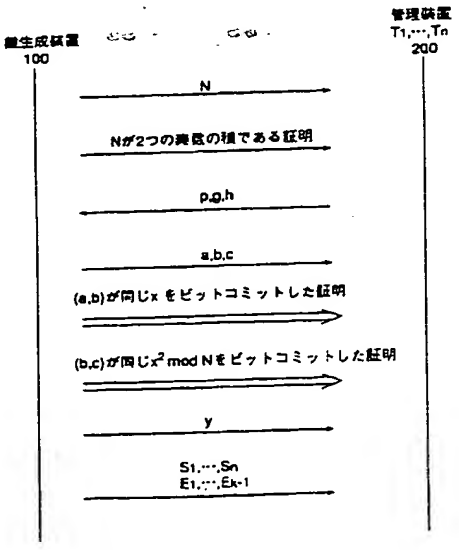


図２

1

【特許請求の範囲】

【請求項１】　鍵生成者の鍵生成装置と、管理者Ｔ$_i$（ｉ＝１，…，ｎ）の管理装置とにより、素因数分解問題に基づく公開鍵暗号方式の秘密鍵を複数に分散して上記管理装置に登録し後に復元する秘密鍵分散管理方法であって、

鍵生成者は鍵生成装置により秘密鍵（Ｐ，Ｑ）と公開鍵Ｎ＝ＰＱを生成し、Ｎを管理者Ｔ$_i$（ｉ＝１，…，ｎ）の管理装置に送り管理装置は単独又は複数共同で素数ｐならびに位数がＮの元ｇを定めて鍵生成装置へ送り、

鍵生成装置はｙ，$a=F_g(x) \bmod p$ を管理装置に送るとともに、Ｎが２つの素数の積であることならびにｘとｙが一定の関係を満足することをｘ，Ｐ，Ｑの値を秘密にして管理装置との間の情報交換により、管理者に証明し、

鍵生成装置はｘをShamirの多項式補間法を用いてｎ個の値$s_1,\cdots, s_n$に分散し、s_iを管理者T_iの管理装置に送ると同時に、s_iが正しく分散された値であることを証明する情報を送り、

管理装置はそれらの正当性を検証し、正しければそれらを受けとり保管し、秘密鍵（Ｐ，Ｑ）を復元する必要があるときにはｎ人の管理者の中でｋ人が協力して、各管理者の分散情報をその管理装置より鍵生成装置に供給し、鍵生成装置はこれら供給された分散情報よりｘの値を復元し、それとｙの値よりＮの素因数Ｐ，Ｑを計算することを特徴とする秘密鍵分散管理方法。

【請求項２】　鍵生成者の鍵生成装置と、管理者Ｔ$_i$（ｉ＝１，…，ｎ）の管理装置によりＲＳＡ公開鍵暗号方式の秘密鍵を複数に分散して上記管理装置に登録し、後に復元する秘密鍵分散管理方法であって、

鍵生成者は鍵生成装置によりＲＳＡ公開鍵暗号方式の秘密鍵ｄと公開鍵（Ｎ，ｅ）を生成し、（Ｎ，ｅ）を管理者T_i（ｉ＝１，…，ｎ）の管理装置に送り、

管理者は単独又は複数共同で管理装置により乱数Ｘを生成すると共に$G=X^e \bmod N$を計算し、また素数ｐならびにパラメータｇを生成し、Ｘ，ｐ，ｇを鍵生成装置に送り、

鍵生成装置は$a=F_g(d) \bmod p$を管理装置に送るとともに、鍵生成者はａと$X=F'_G(d) \bmod N$がそれぞれ同じｄを暗号化していることをｄの値を秘密にしたまま管理者鍵生成装置と管理装置との情報交換により証明し、

鍵生成装置はｄをShamirの多項式補間法を用いてｎ個の値$s_1,\cdots, s_n$に分散し、s_iを管理者T_iの管理装置に送ると同時に、s_iが正しく分散された値であることを証明する情報を送り、

管理装置はそれらの正当性を検証し、正しければそれらを受けとり保管し、

秘密鍵ｄを復元する必要があるときにはｎ人の管理者の中でｋ人が協力して、各管理者の分散情報をその管理装置より鍵生成装置へ供給し、鍵生成装置は供給された分散情報よりｄの値を復元することを特徴とする秘密鍵分散管理方法。

2

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】この発明は、公開鍵暗号の秘密鍵を分散して登録しておき、紛失時やその他の理由により秘密鍵を復元する必要のあるときに、ある一定以上の数の分散登録情報を集めることにより秘密鍵を復元する秘密鍵分散管理方法に関するものである。

【０００２】

【従来の技術】従来、単に情報を分散管理し復元する方式はSharmir の多項式補間法（“Howto Share a Secret ”, Comm. Assoc. Comput. Mach.,vol.22, no.11, pp. 612-613 (Nov.1979)）があるが、分散情報が正しい情報であることを証明する手段がないため、そのままでは秘密鍵を分散して登録する手段には使えない。分散情報が正しい情報であることを証明する方式としては、PedersenによるＶＳＳと呼ばれる方法がある（“Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing”, Proc.of Crypto'91, LNCS 576, Springer-Verlag, pp.129-140(1992)）。しかし、この方法ではElGamal 法などの離散対数問題系の公開鍵暗号には適用できるが、ＲＳＡ法などの素因数分解系の公開鍵暗号には適用できない。

【０００３】一方、Micaliにより（“Fair Public-Key Cryptosystems ”, Proc.of Crypto'92, LNCS, Springer-Verlag, pp.113-138(1993)）ＲＳＡ法などに適用できる秘密鍵の分散管理／復元方式が提案されているが、管理者が全員協力しないと秘密鍵を復元できない（つまり、一人でも協力しないと秘密鍵を復元できなくなる）。

【０００４】

【発明が解決しようとする課題】この発明の目的は、素因数分解系の公開鍵暗号に対して秘密鍵を分散して登録しておき、ある一定以上の数の分散登録情報を集めることにより秘密鍵を復元する方法を実現することにある。

【０００５】

【課題を解決するための手段】この発明では、秘密鍵もしくは秘密鍵と等価な秘密情報を例えばビットコミットメントとよばれる手法で暗号化し、そこで暗号化された秘密情報が正しい情報であることを証明する効率的な方法を開発した。一旦正しい秘密情報を暗号化したビットコミットメントが得られると、前述したPedersenの手法を用いて、分散情報が正しい情報であることを証明することが可能である。

【０００６】請求項１では、ｙを公開しておき、公開鍵Ｎを素因数分解するための秘密情報として$x^4 \equiv y^2 \pmod N$であるようなｘを用いる。請求項２では、ＲＳＡ暗号の秘密鍵ｄをビットコミットメントの対象とす

る。

【０００７】

【発明の実施の形態】以下では、この発明の請求項１の一実施例について説明する。図１はこの発明の全体構成を示す。鍵生成者の装置（鍵生成装置）１００は、管理者T_iの装置（管理装置）２００ｉ（ｉ＝１，２，…，ｎ）とそれぞれ、通信路３００ｉを介して結合されているとする。図２にこの発明の通信シーケンス例を示し、以下、それぞれ図３に鍵生成装置１００の機能構成例を、図４に管理装置２００ｉの機能構成例を示す。

【０００８】以下、分散情報の登録ならびに秘密鍵を回復する手順を示す。

１．鍵生成者は鍵生成装置１００内の鍵生成器１０１を用いて、秘密鍵（Ｐ，Ｑ）と公開鍵Ｎ＝ＰＱを生成し、Ｎを管理者T_iの管理装置２００ｉ（ｉ＝１，…，ｎ）に送る。ここでヤコビ記号の値が（－１／Ｎ）＝１とする。これはＮを素因数Ｐ，Ｑに分解できる条件である。

【０００９】２．鍵生成装置１００の剰余演算器１０２を用いてＮが２つの素数の積であることを証明する情報を生成し、それを管理装置２００ｉに送る。このような情報の作成方法は、Micaliの方法（“Fair Public-Key Cryptosystems”，Proc.ofCyrpto'92, LNCS, Springer-Verlag, pp.113-138(1993)）の中で示されている。

３．管理装置２００ｉは単独又は共同でパラメータ生成器２０１を用いて素数ｐならびに位数がＮの元ｇ，ｈを定め、それを鍵生成装置１００に送る。ここで、ビットコミットメント関数ＢＣを以下のように定義する。

【００１０】$BC_g(x, r) = g^x h^r \bmod p$４．鍵生成装置１００は乱数生成器１０３を用いて乱数ｘならびに適当な値r_1, r_2, r_3を定め、剰余演算器１０４を用いて次のａ，ｂ，ｃを計算し、管理装置２００ｉに送る。

$a = BC_g(x, r_1)$

$b = BC_a(x, r_2) = BC_g(x^2 \bmod N, r_1 x + r_2 \bmod N)$

$c = BC_b(x^2 \bmod N, r_3) = BC_g(x^4 \bmod N, r_1 x_3 + r_2 x^2 + r_3 \bmod N)$

５．次に、鍵生成装置１００は$a = BC_g(x, r_1)$と$b = BC_a(x, r_2)$が同じｘをビットコミットしたものであることを以下の手順で証明する。

【００１１】（ａ）鍵生成装置１００は乱数生成器１０３を用いてｖ，ｗ，ｗ′を生成し、さらに剰余演算器１０５を用いて

$u = BC_g(v, w)$，$u' = BC_a(v, w')$

を生成し、（ｕ，ｕ′）を管理装置２００ｉに送る。

（ｂ）管理装置２００ｉは乱数生成器２０２を用いてＲを定め、それを鍵生成装置１００に送る。

【００１２】（ｃ）鍵生成装置１００は剰余演算器１０６を用いて

$z = v + xR \bmod N$，$t = w + r_1 R \bmod N$，$t' = w' + r_2 R \bmod N$

を計算し、（ｚ，ｔ，ｔ′）を管理装置２００ｉに送る。

（ｄ）管理装置２００ｉは以下の式が成立するかどうかを剰余演算器２０３ならびに比較器２０４を用いて検証する。

【００１３】$BC_g(z,t) = u a^R \bmod p$，$BC_a(z, t') = u' b^R \bmod p$

６．鍵生成装置１００は同様に

$b = BC_g(x^2 \bmod N, r_1 x + r_2 \bmod N)$

$c = BC_b(x^2 \bmod N, r_3)$

が同じ$x^2 \bmod N$をビットコミットしたものであることを上と同じ手順で証明する。

【００１４】７．鍵生成装置１００は剰余演算器１０７を用いて$y^2 \equiv x^4 \pmod N$およびヤコビ記号の値が（ｙ／Ｎ）＝－１を満足するｙを計算し管理装置２００ｉに送る。ここでｙはｘと$y^2 = x^4 \pmod N$の関係にあることが前記第５ステップと第６ステップの両証明により証明されており、かつ（ｙ／Ｎ）＝－１，ｘ，ｙが与えられるとＮは素因数Ｐ，Ｑに分解できる条件である。

【００１５】８．鍵生成装置１００はｘをSharmir の多項式補間法により剰余演算器１０８を用いてｎ個の値$s_1, \cdots, s_n$に分散し、s_iを管理者T_iの管理装置２００ｉに送る。さらに、$a = BC_g(x, r_1)$の値を利用して、Pedersenの方法により剰余演算器１０８を用いてs_iが正しく分散された値であることを証明する情報$E_1, \cdots, E_{k-1}$を送る。ｋ－１は前記多項式補間法に用いる多項式の最低の次数であり、$k \leq n$である。

【００１６】９．管理装置２００ｉは剰余演算器２０５および比較器２０６を用いてそれらの正当性を検証し、正しければそれらを受けとり保管する。

１０．秘密鍵（Ｐ，Ｑ）を復元する必要があるときにはｎ人の管理者の中でｋ人が協力して、各管理者の分散情報s_iを持ち寄りそれらよりShamirの方法により剰余演算器１０９を用いてｘの値を復元する。さらに、ＧＣＤ演算器１１０を用いてＧＣＤ演算器１１０を用いて、$x^2 - y$とＮとの最大公約数を求めると、Ｐ又はＱが求まり、これより、Ｐ，Ｑの両者を求めることができる。つまり、いま$m = x^2 \pmod N$、ヤコビ記号の値が（ｍ／Ｎ）＝１とすると、$m^2 = y^2 = x^4 \pmod N$であり、かつ（１／Ｎ）＝－１，（ｙ／Ｎ）＝－１の条件がある場合は、（ｚ－ｙ），つまり$(x^2 - y)$はＮの素因数で割り切れることが数学的に知られている。よって$x^2 - y$とＮとの最大公約数を求めればＮの１つの素因数が得られることになる。

【００１７】次に、この発明の請求項２の一実施例について説明する。図５にこの発明の通信シーケンス例を示し、以下、それぞれ、図６に鍵生成装置１００の構成例を、図７に管理装置２００ｉの構成例を示す。以下、分

散情報の登録ならびに秘密鍵を回復する手順を示す。
１．鍵生成装置１００は鍵生成器１０１を用いて、ＲＳＡ暗号の秘密鍵ｄと公開鍵（Ｎ，ｅ）を生成し、（Ｎ，ｅ）を管理者T_i（ｉ＝１，…，ｎ）の管理装置２００ｉに送る。
【００１８】２．管理装置２００ｉは単独又は共同でパラメータ生成器２０１を用いて素数（ｐ，ｑ）ならびに位数がｑの元ｇ，ｈを定め、それを鍵生成装置１００に送る。なお、ｑ｜ｐ－１（ｑはｐ－１の約数）とする。ここで、ビットコミットメント関数ＢＣを以下のように定義する。$BC_g(x, r) = g^x h^r \bmod p$
さらに管理装置２００ｉは乱数生成器２０２を用いてＸを生成し、剰余演算器２０３を用いて$G = X^e \bmod N$を計算する。
【００１９】管理装置２００ｉはＸならびに（ｐ，ｑ，ｇ，ｈ）を鍵生成者に送る。ここで、$BC_g(s, r) = g^s h^r \bmod p$，$BC_G(s) = G^s \bmod N$とする。
３．鍵生成装置１００は乱数生成器１０３を用いて適当な値r_1を定め、剰余演算器１０２を用いて
$a = BC_g(d, r_1)$
を計算するとともに
$b = BC_G(d) = X$
とする。
【００２０】４．次に、鍵生成装置１００は$a = BC_g(d, r_1)$と$b = BC_G(d) = X$が同じｄをビットコミットしたものであることを以下の手順で証明する。
（ａ）鍵生成装置１００は乱数生成器１０３を用いてｖ，ｗを生成し、さらに剰余演算器１０４を用いて
$u = BC_g(v, w)$，$u' = BC_G(v)$
を生成し、（ｕ，ｕ′）を管理装置２００ｉに送る。なおｕ′の計算に用いるＧはＸ，ｅ，Ｎにより計算する。
【００２１】（ｂ）管理装置２００ｉは乱数生成器２０２を用いてＲを定め、それを鍵生成装置１００に送る。
（ｃ）鍵生成装置１００は、剰余演算器１０５を用いて
$z = v + dR$，　$t = w + r_1 R \bmod q$
を計算し、（ｚ，ｔ）を管理装置２００ｉに送る。
【００２２】（ｄ）管理装置２００ｉは以下の式が成立するかどうかを剰余演算器２０４ならびに比較器２０５を用いて検証する。
$BC_g(z,t) = u a^R \bmod p$，$BC_G(z) = u' b^R \bmod N$
さらに比較器２０５を用いてｚがｖ，Ｒ，ｄのサイズ（ビット数）で決るビット数以下であることを確認する。
【００２３】５．鍵生成装置１００はｄをShamirの多項式補間法により剰余演算器１０６を用いてｎ個の値$s_1, \cdots, s_n$に分散し、s_iを管理者T_iの管理装置２００ｉに送る。さらに、$a = BC_g(d, r_1)$の値を利用して、Pedersenの方法により剰余演算器１０６を用いてs_iが正しく分散された値であることを証明する情報$E_1, \cdots, E_{k-1}$を送る。
【００２４】６．管理装置２００ｉは剰余演算器２０６および比較器２０７を用いてそれらの正当性を検証し、正しければそれらを受けとり保管する。
７．秘密鍵ｄを復元する必要があるときにはｎ人の管理者の中でｋ人が協力して、各管理者の分散情報を持ち寄りそれらよりShamirの方法により剰余演算器１０７を用いてｄの値を復元する。
【００２５】上述では請求項１の発明で$a = F_g(x) \bmod p$として$g^x h^r \bmod p$を用いたが$g^x \bmod p$でもよい。同様に請求項２の発明でも$a = F_g(d) \bmod p$として$g_d \bmod p$でもよい。
【００２６】
【発明の効果】請求項１の発明では、素因数分解系の公開鍵暗号であるＲＳＡ法やRabin 法に対して秘密鍵（Ｐ，Ｑ）に関連する秘密情報ｘを分散して登録しておき、しかも各管理者は自分に送られてきた分散情報が正しいことを確認できる。従って、一定人数（ｋ人）の管理者が協力すれば必ず秘密鍵（Ｐ，Ｑ）を復元できる。また、同様に請求項２の発明では、ＲＳＡ法に秘密鍵ｄを分散管理し一定人数（ｋ人）の管理者が協力すれば必ず秘密鍵ｄを復元できる。
【図面の簡単な説明】
【図１】この発明が適用されるシステムの構成を示すブロック図。
【図２】請求項１の発明の実施例における鍵生成装置と管理装置間で行う処理シーケンスを示す図。
【図３】図２の実施例における鍵生成装置１００の機能構成を示すブロック図。
【図４】図２の実施例における管理装置２００ｉの機能構成を示すブロック図。
【図５】請求項２の発明の実施例における鍵生成装置と管理装置間で行う処理シーケンスを示す図。
【図６】図５の実施例における鍵生成装置１００の機能構成を示すブロック図。
【図７】図５の実施例における管理装置２００ｉの機能構成を示すブロック図。

【図１】

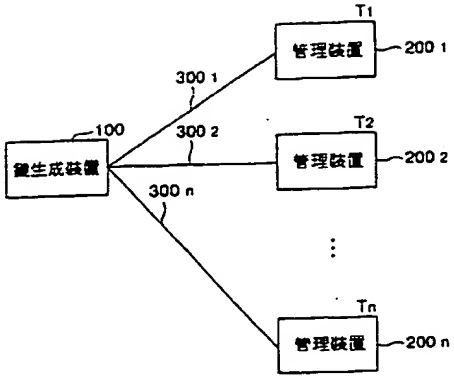


図 1

【図２】

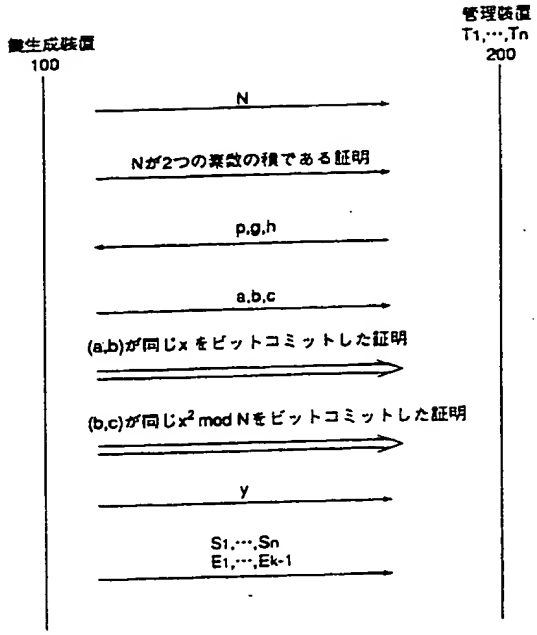


図 2

【図４】

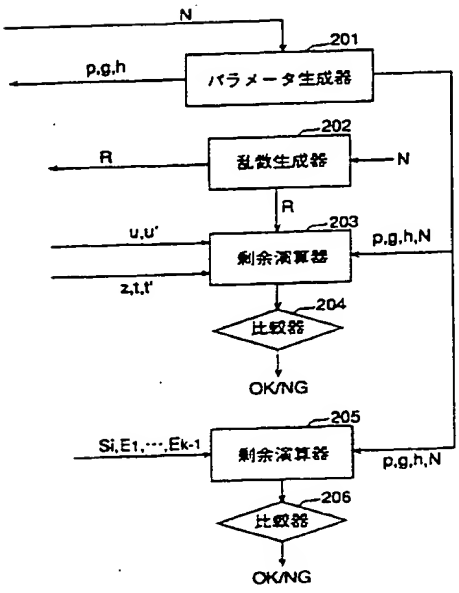


図 4

【図３】

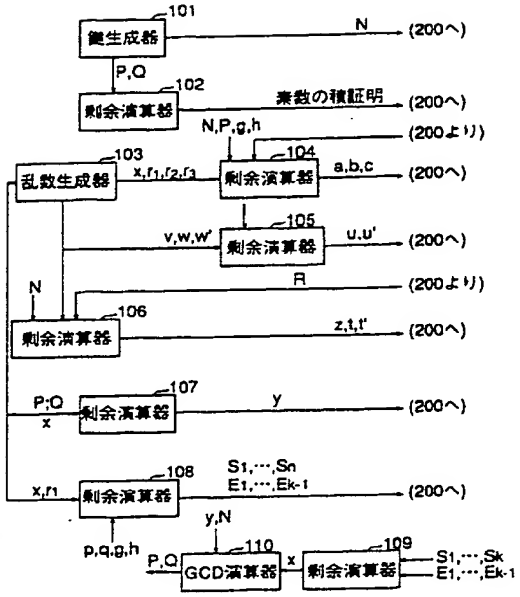

101
素生成器
N
(200へ)
P,Q
102
剰余演算器
素数の積証明
(200へ)
N,P,g,h
(200より)
103
乱数生成器
x,r1,r2,r3
104
剰余演算器
a,b,c
(200へ)
105
剰余演算器
v,w,w'
u,u'
(200へ)
R
(200より)
N
106
剰余演算器
z,t,t'
(200へ)
107
剰余演算器
P;Q
x
y
(200へ)
108
剰余演算器
x,r1
S1,…,Sn
E1,…,Ek-1
(200へ)
y,N
p,q,g,h
110
GCD演算器
P,Q
x
109
剰余演算器
S1,…,Sk
E1,…,Ek-1

図　3

【図５】

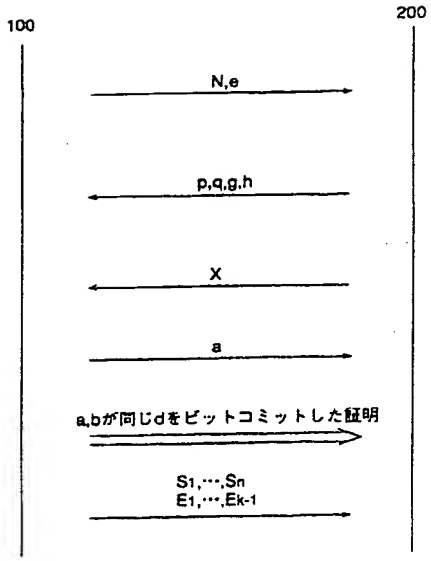

100
200
N,e
p,q,g,h
X
a
a,bが同じdをビットコミットした証明
S1,…,Sn
E1,…,Ek-1

図　5

【図６】

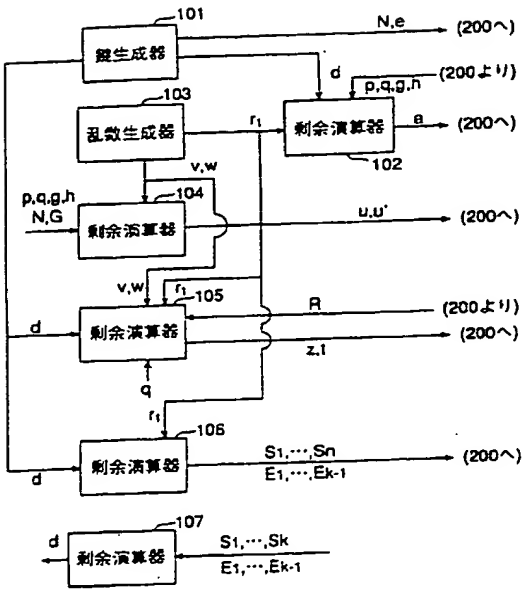

101
乱数生成器
N,e
(200へ)
d
p,q,g,h
(200より)
103
乱数生成器
r1
剰余演算器
e
(200へ)
102
v,w
p,q,g,h
N,G
104
剰余演算器
u,u'
(200へ)
v,w
r1
105
R
(200より)
d
剰余演算器
(200へ)
z,t
q
r1
106
剰余演算器
S1,…,Sn
E1,…,Ek-1
(200へ)
d
107
d
剰余演算器
S1,…,Sk
E1,…,Ek-1

図　6

【図７】

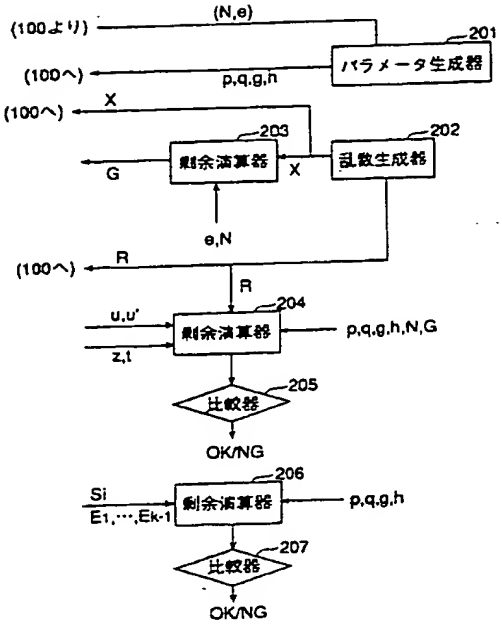

(100より)
(N,e)
201
パラメータ生成器
(100へ)
p,q,g,h
(100へ)
X
203
剰余演算器
202
乱数生成器
G
X
e,N
(100へ)
R
R
u,u'
204
剰余演算器
p,q,g,h,N,G
z,t
205
比較器
OK/NG
Si
E1,…,Ek-1
206
剰余演算器
p,q,g,h
207
比較器
OK/NG

図　7